



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/867,650	05/31/2001	Michael Anthony Sijacic	06502.0338-00000	7983
60667	7590	02/06/2008		
SUN MICROSYSTEMS/FINNEGAN, HENDERSON LLP 901 NEW YORK AVENUE, NW WASHINGTON, DC 20001-4413			EXAMINER BULLOCK JR, LEWIS ALEXANDER	
			ART UNIT 2195	PAPER NUMBER
			MAIL DATE 02/06/2008	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

09/867,650

Applicant(s)

SIJACIC ET AL.

Examiner

Lewis A. Bullock, Jr.

Art Unit

2195

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 07 January 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-34 and 36-91 is/are pending in the application.
- 4a) Of the above claim(s) 20, 37, 38, 59 and 85 is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-19, 21-34, 36, 39-58, 60-84 and 86-91 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/ are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- ☐ Notice of Informal Patent Application
- ☐ Other: _____

Election/Restrictions

1. Applicant's election without traverse of Group I, claims 1-19, 21-34, 36, 39-58, 60-84, and 86-91 in the reply filed on September 11, 2007 is acknowledged.
2. Claims 20, 37, 38, 59, and 85 are withdrawn from further consideration pursuant to 37 CFR 1.142(b) as being drawn to a nonelected invention, there being no allowable generic or linking claim. Election was made **without** traverse in the reply filed on September 11, 2007.
3. This application contains claims 20, 37, 38, 59 and 85 drawn to an invention nonelected without traverse in the reply filed on September 11, 2007. A complete reply to the final rejection must include cancellation of nonelected claims or other appropriate action (37 CFR 1.144). See MPEP § 821.01.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-19, 27-34, 36, 40-58 and 66-84 are rejected under 35 U.S.C. 102(e) as being anticipated by CHONG (U.S. Patent Application Publication 2002/0184610).

As to claim 1, CHONG teaches a method for creating an activity (component / object / workflow step) within a process management system (development system / workflow environment), comprising: receiving first data reflecting a class file that implements an interface (adapter - class); receiving second data reflecting a data representation file that specifies environment, input parameters, and output parameters for the class (imported media / variables / via a functional black-box interface that allows variables to be passed in and out of the component); packaging the first and second data (via receiving the classes and variable information packaged in a Jar file) (via componentization) (pg. 29, paragraph 431-434); and associating the packaged data with an activity (via all information are stored in a Jar file) that may be used in an automated workflow process (workflow) (via dragging and dropping the icon representing the component to a diagram of the workflow) (pg. 13, paragraph 257 – 258; pg. 14, paragraph 262) to access information external to the process management system (via using the adapters to access the external data sources and manipulate the external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354).

As to claim 12, CHONG teaches a method for implementing a custom activity (component / object / workflow step) within a process management environment (development system / workflow environment), comprising: defining a file associated with a custom activity (via the user creating a component and the class of the component via componentization) (pg. 29, paragraph 430-434); assigning a visual

representation associated with the custom activity (via packaging the visual icon via componentization) (pg. 29, paragraph 430-434); receiving an indication (via drag and drop operation) reflecting implementation of the custom activity in a workflow process based on a position of the visual representation in a process map (workflow diagram) representing the workflow process (via dragging and dropping the icon representing the component to a diagram of the workflow and linking the instantiated component) (pg. 13, paragraph 257 – 258; pg. 14, paragraph 262); and invoking the file (via sending and loading the file and allowing the component to be used in an application) (pg. 30, paragraph 433) wherein the file includes a class file that implements an interface (adapter - class) and a data representation file that specifies an environment, input parameters, and output parameters for the class (imported media / variables / via a functional black-box interface that allows variables to be passed in and out of the component).

As to claim 14, CHONG teaches a method for creating and defining a custom activity (component / object / workflow step) within a process management system (development system / workflow environment), comprising: creating at least one file defining properties (various attributes of the component, i.e. items i-viii) associated with the custom activity (pg. 29, paragraph 430-434); and defining a model (visual icon) associated with the custom activity (via packaging the visual icon via componentization) (pg. 29, paragraph 430-434), wherein the custom activity may be used to access information external to the process manager system (via using the adapters of the

component to access the external data sources and manipulate the external data sources) wherein the file includes a class file that implements an interface (adapter - class) and a data representation file that specifies an environment, input parameters, and output parameters for the class (imported media / variables / via a functional black-box interface that allows variables to be passed in and out of the component) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354).

As to claim 27, CHONG teaches a memory for storing data for access by a process (application) being executed by a processor, the memory comprising: a structure defining a class file that implements an interface (adapter - class) and a data representation file that specifies an environment, input parameters, and output parameters (imported media / variables / via a functional black-box interface that allows variables to be passed in and out of the component), packaging the files (via receiving the classes and variable information packaged in a Jar file) (via componentization) (pg. 29, paragraph 431-434), assigning an icon (visual icon) representing the packaged files (component) (via the visual icon is part of the Jar file) (pg. 29, paragraph 431-434), and associating the icon with an activity that performs processes defined by the class and data representation files (via drag and dropping the component into the workflow diagram by manipulating the visual icon) (pg. 13-14, paragraph 257-259).

As to claim 30, CHONG teaches a memory for storing data for access by a process (application) being executed by a processor, the memory comprising: a

structure for maintaining an identity of a custom activity (component / object / workflow step) associated with a class file (adapter - class) and a data representation file (imported media / variables / via a functional black-box interface that allows variables to be passed in and out of the component), parameters associated with the custom activity (parameters), a first hashtable reflecting data values to be used as input argument in a method, and a second hashtable reflecting output arguments of the method (via the functional black-box interface) (pg. 29, paragraph 430-434) wherein the data values of the first hashtable and the output arguments of the second hashtable are consistent with one or more data types specified within the data representation file (via the data representation file includes the black-box interface that allows variables to be passed in and out of the component such that it displays a mapping panel that allows for the assigning of variables going into the component and out of the component back to the controller) (see pg. 51, paragraph 0662-0663, see also figure 112 wherein the input column and the output column are the hashtables).

As to claim 31, CHONG teaches a memory for storing data for access by a process (application) being executed by a processor, the memory comprising: a structure for defining a value of a parameter associated with an input hashtable (variable to be passed in via the black-box interface / data bindings) (pg. 29, paragraph 430-434; pg. 24, paragraph 353-357), mapping a value of a parameter associated with an output hashtable (variable to be passed out via the blackbox interface / data bindings) (pg. 29, paragraph 430-434; pg. 24, paragraph 353-357), and defining a user

interface associated with a custom activity (visual icon representing the component), associated with a class file (adapter - class) and a data representation file (imported media / variables / via a functional black-box interface that allows variables to be passed in and out of the component) (pg. 29, paragraph 430-434) that performs a process based on the values of the parameters in the input and output hashtables (via using the adapters to access external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354) wherein the data values of the first hashtable and the output arguments of the second hashtable are consistent with one or more data types specified within the data representation file (via the data representation file includes the black-box interface that allows variables to be passed in and out of the component such that it displays a mapping panel that allows for the assigning of variables going into the component and out of the component back to the controller) (see pg. 51, paragraph 0662-0663, see also figure 112 wherein the input column and the output column are the hashtables).

As to claims 32 and 33, CHONG teaches a memory for storing data associated with a custom activity for access by a process (application) being executed by a processor the memory comprising: a structure (workflow) specifying an input tag that obtains a value for an input hashtable to be used as an argument in a method (via the setvariableaction tag / modelinterface tag / modelvardef tag / databindinginterface tag / databinding tag wherein a data binding identifies both the input variables and the output variables), specifying an output tag that specify parameters that define what to do with

parameters in an output hashtable including output arguments associated with the method (via performing any action tag / databindinginterface tag / databinding tag wherein a data binding identifies both the input variables and the output variables), and specifying design tags that define a user interface associated with the custom activity associated with a class file (adapter - class) and a data representation file (imported media / variables / via a functional black-box interface that allows variables to be passed in and out of the component) (listview tag / devicebasedview tag) (pg. 16, paragraph 284 – 297; pg. 24, paragraph 351-357) wherein the data values of the first hashtable and the output arguments of the second hashtable are consistent with one or more data types specified within the data representation file (via the data representation file includes the black-box interface that allows variables to be passed in and out of the component such that it displays a mapping panel that allows for the assigning of variables going into the component and out of the component back to the controller) (see pg. 51, paragraph 0662-0663, see also figure 112 wherein the input column and the output column are the hashtables).

As to claim 34, CHONG teaches a memory for storing data for access by a process (application) being executed by a processor, the memory comprising: a structure defining a custom activity (component / object / workflow step) implemented in a process management system (development system / workflow environment) by defining a package (component) (via componentization) (pg. 29, paragraph 431-434) for importing packages external to the process management system (via using the

adapters to access external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354), defining an init method for defining initialization tasks associated with the custom activity (pre-action tasks on the component) (pg. 15, paragraph 267-271; pg. 12, paragraph 231-234), and defining a perform method for executing tasks associated with the custom activity (action tasks of the component) (pg. 15, paragraph 267-271; pg. 12, paragraph 231-234) wherein the first hashtable includes values corresponding to data fields and the second hashtable includes values to be placed in fields (via the data representation file includes the black-box interface that allows variables to be passed in and out of the component such that it displays a mapping panel that allows for the assigning of variables going into the component and out of the component back to the controller) (see pg. 51, paragraph 0662-0663, see also figure 112 wherein the input column and the output column are the hashtables). It is inherent to the teachings of CHONG that since the component executes using the variables that it is associated with the variables.

As to claim 2, CHONG teaches the data representation file includes a section that determines the appearance of a representation reflecting the activity (visual icon) (pg. 29, paragraph 431-434).

As to claim 3, CHONG teaches the class file includes a method that is configured (via the adapter bindings and the functional black box interface) to obtain a value of a

parameter defined in the data representation file (pg. 29, paragraph 431-434; pg. 24, paragraph 351-357).

As to claim 4, CHONG teaches receiving data that defines a package for the class file; and receiving data that defines methods that retrieve and set values to variables to be used by the activity (via componentizing the behavior of the component and instruction on how to assemble the resources of the component into the application (pg. 29, paragraph 434-434).

As to claim 5, CHONG teaches receiving data that reflects a method that defines variables that are constant across all instances of the activity (via the black-box interface / data bindings that defines both input and output variables) (pg. 29, paragraph 430-434; pg. 24, paragraph 353-357; pg. 30, paragraph 433).

As to claim 6, CHONG teaches the method is associated with an input hashtable to define values of a variable used by the activity (variable to be passed in via the black-box interface / data bindings) (pg. 29, paragraph 430-434; pg. 24, paragraph 353-357; pg. 30, paragraph 433).

As to claim 7, CHONG teaches receiving data (for componentization or at the receiving node of the Jar file) reflecting a method that defines values for variables in a first hashtable and retrieves values for variables from a second hashtable (variable to

be passed in and out via the black-box interface / data bindings) (pg. 29, paragraph 430-434; pg. 24, paragraph 353-357; pg. 30, paragraph 433).

As to claim 8, CHONG teaches receiving data reflecting a method that releases resources used by an application that implements the activity when the application is unloaded from the process management system (via the delete action) (pg. 14-15, paragraph 265).

As to claim 9, CHONG teaches receiving data reflecting a first section that defines a type and name of the class file (adapter); receiving data reflecting a second section that defines parameters with values that remain constant within all instances of the activity (via the variables or data bindings provided); receiving data reflecting a third section that sets values for selected parameters within a first hashtable (via an action operation on the variables); receiving data reflecting a fourth section that defines what to do with parameters included in a second hashtable (via instruction on how to assemble the resources in the component into the application); and receiving data reflecting a fifth section associated with a visual representation associated with the activity (visual icon of the component) (pg. 29, paragraph 431-435).

As to claim 10, CHONG teaches packaging the first and second data into one of a JAR file or a ZIP file (pg. 29, paragraph 431).

As to claim 11, CHONG teaches locating the packaged data; and receiving data reflecting a visual representation (visual icon) that corresponds to the packaged files (pg. 29, paragraph 431-434).

As to claim 13, CHONG teaches the file is an archive file and includes the visual representation (pg. 29, paragraph 431-434).

As to claim 15, CHONG teaches the model is an image reflecting the custom activity (visual icon) (pg. 29, paragraph 431-434).

As to claim 16, CHONG teaches packaging the file and model into an archive file (pg. 29, paragraph 431-434).

As to claim 17, CHONG teaches associating the custom activity with a workflow process managed by the process management system (via dragging and dropping the component via its visual icon into the workflow diagram) (pg. 13, paragraph 257 – 258; pg. 14, paragraph 262).

As to claim 18, CHONG teaches determining a position of the model in a visual process map (workflow diagram) reflecting the workflow process; and invoking the custom activity in the workflow process based on the determination (via linking the

components of the workflow diagram) (pg. 13, paragraph 257 – 258; pg. 14, paragraph 262).

As to claim 19, CHONG teaches the at least one file includes a Java class file (object oriented class) (pg. 7, paragraph 149; pg. 24, paragraph 351-352; pg. 25, paragraph 369) and an XML description file (pg. 8, paragraph 176; pg. 19, paragraph 325).

As to claims 28 and 29, CHONG teaches the data representation is an XML description file (pg. 8, paragraph 176; pg. 19, paragraph 325).

As to claim 36, CHONG teaches the custom activity may have a plurality of instances and wherein the init method defines an association with resources external to the process management system and are shared by all instances of the custom activity (via a component is a container of other components and the components have instructions on how to assemble the resources in the component into the application using that component) (pg. 29, paragraph 431-434).

As to claims 40-58, reference is made to a computer readable medium that corresponds to the method of claims 1-19 and is therefore met by the rejection of claims 1-19 above.

As to claims 66-84, reference is made to a system that corresponds to the method of claims 1-19 and is therefore met by the rejection of claims 1-26 above.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 21-26, 39, 60-65 and 86-91 are rejected under 35 U.S.C. 103(a) as being unpatentable over CHONG (U.S. Patent Application Publication 2002/0184610).

As to claim 21, CHONG teaches a method for creating a custom activity (component / object / workflow step) in a process management system (development system / object / workflow step), the custom activity exchanging information with resources external to the process management system (via using the adapters of the component to access the external data sources and manipulate the external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354), comprising: receiving a first file reflecting a class file that implements an interface (adapter - class) and a second file; and archiving the files in an archive file (via receiving the classes and variable information packaged in a Jar file) (via componentization) (pg. 29, paragraph 431-434) such that when the custom activity is activated the archived files are accessed and executed (via sending and loading the Jar file and allowing the component to be used when dragged into the workflow diagram in an application) (pg.

30, paragraph 433). CHONG also teaches that the data / files of the component are stored in a Jar file or project, which is a directory of the system (pg. 30, paragraph 0434 – 0435). Official Notice is taken in that it is well known in the art that a files within a directory have the same root name, but different file names within that directory root name and therefore obvious to one of ordinary skill in the art that the files / data of CHONG which are stored in the same directory / initial JAR file have the same root name. CHONG also teaches making a copy of the JAR file / project such that the componentization model sends a copy to be later associated into an application (pg. 30, paragraph 0435-0437). It is well known in the art and would be obvious to one of ordinary skill in the art that typical file copy procedures would ensure the copy is correct (error free) or that all information is copied before sending the copy over and therefore obvious in view of CHONG that the copy is verified to reflect the package structure before sending the JAR file to be later associated into the application.

As to claim 39, CHONG teaches a system for creating an implementing a custom activity (component / object / workflow step) in a process management environment (development system / workflow environment), comprising: a processor; and a memory containing instructions executable by the processor (pg. 8, paragraph 177-178) to: receive a first file reflecting a class file that implements an interface (adapter - class) and defining with an interface with a package external to the process management system (external data source); receive a second file defining parameters (variables) that the first file uses; archive the first and second file in an archive file (via receiving the

classes and variable information packaged in a Jar file) (via componentization) (pg. 29, paragraph 431-434); and invoke the first and second file based on a manipulation of an image reflecting the custom activity in a visual process map (workflow diagram) reflecting an automated workflow process (via dragging and dropping the icon representing the component to a diagram of the workflow) (pg. 13, paragraph 257 – 258; pg. 14, paragraph 262). CHONG also teaches that the data / files of the component are stored in a Jar file or project, which is a directory of the system (pg. 30, paragraph 0434 – 0435). Official Notice is taken in that it is well known in the art that a files within a directory have the same root name, but different file names within that directory root name and therefore obvious to one of ordinary skill in the art that the files / data of CHONG which are stored in the same directory / initial JAR file have the same root name. CHONG also teaches making a copy of the JAR file / project such that the componentization model sends a copy to be later associated into an application (pg. 30, paragraph 0435-0437). It is well known in the art and would be obvious to one of ordinary skill in the art that typical file copy procedures would ensure the copy is correct (error free) or that all information is copied before sending the copy over and therefore obvious in view of CHONG that the copy is verified to reflect the package structure before sending the JAR file to be later associated into the application.

As to claim 22, CHONG teaches receiving package information associated with the first file (adapters) that implement packages external to the process management system (via using the adapters to access the external data sources and manipulate the

external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354).

As to claim 23, CHONG teaches receiving data that interacts with parameters associated with a hashtable defined in the second file (via performing functions to acquire the in variables for the adapter) (pg. 24, paragraph 353-357).

As to claim 24, CHONG teaches receiving data associated with the second file that defines at least one hashtable used by the first file (via defining or retrieving the in variables for the adapter) (pg. 24, paragraph 353-357).

As to claim 25, CHONG teaches the first file reflects a class file (object oriented class) (pg. 7, paragraph 149; pg. 24, paragraph 351-352; pg. 25, paragraph 369) and the second file reflects an XML file (pg. 8, paragraph 176; pg. 19, paragraph 325).

As to claim 26, CHONG teaches archiving the files in an archive file consisting of one of a JAR file and a ZIP file (pg. 29, paragraph 431-434).

As to claims 60-65, reference is made to a computer readable medium that corresponds to the method of claims 21-26 and is therefore met by the rejection of claims 21-26 above.

As to claims 86-91, reference is made to a system that corresponds to the method of claims 21-26 and is therefore met by the rejection of claims 21-26 above.

Response to Arguments

8. Applicant's arguments filed January 7, 2008 have been fully considered but they are not persuasive. Applicant argued that Chong does not teach first data and second data because Chong does not teach the first data reflecting a class file that implements an interface and second data reflecting a data representation file that specifies an environment, input parameters, and output parameters for the class. Applicant further states that the interface is implemented by the controller and is kept separate from the class file. The examiner disagrees. Chong states that each component created by the system comprises at least: all media imported into the component including strings, images, audio files, and speech recognition grammars; and a description that defines the behavior of the component (pg. 29, paragraph 0431). These components are stored in an archive file format, one of which is a Java JAR file (pg. 29, paragraph 0431). The componentization process opens the JAR file (which is a collection of multiple files that have been concatenated and indexed so that the files can be later extracted into individual files) and writes each of the components into the JAR file (pg. 30, paragraph 0437). Chong further details an example, wherein one wants to use the currency converter component as part of a larger application (travel porter) (pg. 50, paragraph 0651). Chong teaches the converter component comprises the (i) interaction flow for the converter as seen under the "Controllers" folder; (ii) the back-end data adapters and

bindings, i.e. the SQL adapter and Java classes used for performing the conversion, as seen under the "Data Model " folder; (iii) the presentation layer, i.e. all the views in the project, as seen under the "Views" folder; (iv) all media in the component all under the "Resources" folder; (v) a visual icon representing the component; (vi) a description that defines the behavior of the component; (vii) a functional black-box interface that allows variables to be passed in and out of the component; and (viii) instructions on how to assemble the resources in the component into the application using that component wherein items (i) – (viii) are stored in any suitable archive file format (see pg. 50-51, paragraphs 0652-0661). Because the componentization process, opens an archive file and inserts the individual files (directories of information) into the file, the teaching of Chong teaches receiving a first data and a second data. In addition, since items ii and iv is one of the items that at least indicates a Java class and stored in the Data Model folder and all media used in the component is data stored in a resources folder, Chong at least by these two components teach data reflecting a class file and data reflecting a data representation file. The teachings of Chong at least allows for the directories to be received by a componentization tool to be componentized into an archive file.

Conceivably, item (vi) can also be the class file since a class is a description that defines an object and item (vi) is a description that defines the behavior of the component (object). Therefore, Chong does teach the class file (a description that defines the behavior of the component) that is received and packaged into a JAR file. In addition, Chong teaches the concept of componentization wherein the handle of a JAR file is forwarded to a Project Manager such that it saves all the files of the project

into the JAR file (pg. 30, paragraph 0437). The components that are componentized include, but are not limited to business logic, interface standards, service functionality, and network intelligence components, including common visual controls, data control adapters, transaction blocks, protocol Independent Multicast applications and commerce applications (pg. 31, paragraph 0439). At run-time, components used in a project are instantiated as required by the engine by reading the storage manifest of the component and determine how to instantiate the component wherein the run-time behavior of the component is defined by the sum of all it is comprised of (pg. 31, paragraph 0441). Chong discloses that adapters, which according to paragraphs 0352 and 0647 and are part of the elements of the component are essentially classes whose interface is exposed in the GUI. Items (ii) clearly refers to the adapters as recited as back-end adapters. Chong teaches that the packaged data includes back-end adapters and that the component gathers conversion information as input and process the data accordingly (pg. 49, paragraph 0642). Therefore, Chong teaches the componentization tool enabling the receiving of a class implementing an interface since the data adapter is componentized into the packaged. Chong separates the black-box interface from the adapter class. The black-box interface allows variables to be passed in and out of the component. This element corresponds to a data representation file that specifies environment, input parameters and output parameters for the class since it handles the variables for the component whereas the adapter makes up the functionality of the component. Applicant argues that the data adapter in Chong is a pre-built class that provides a mechanism for instantiation of user objects for each particular type of object

and is implemented by the controller and is kept separate from the class field and therefore would not be packaged with other data. The examiner disagrees. The reference is clear that the components that are componentized include adapters which are Java classes whose interface is exposed in the GUI (pg. 49, paragraph 0647; pg. 50, paragraph 0654). The JAR file contains all of the information needed to communicate with and use the component. This will include the adapter and its interface. Since the component and all of its components are packaged and sent in a JAR file for later using, the adapter and its interface which are part of the component are also packaged in the JAR file. Therefore, data and the class file that implements an interface are also packaged since all of the aspects of the components are packaged for reuse.

Applicant further argues that a black-box interface in Chong only allows variables to be passed in and out of the component, this would not constitute the claimed "first hashtable reflecting data values to be used as input argument" and "a second hashtable reflecting output arguments." The examiner disagrees. Applicant has provided no basis for how the mapping is flawed. The use of the interface to allow variables to be passed in would reflect data values to be used as input arguments and the use of the interface to allow variables to be passed out would reflect output arguments. Therefore, the examiner believes that the limitations are met.

Regarding Applicant remaining arguments to the elected claims, the arguments are directed toward similar arguments made above. In response, the examiner provides

support above in showing that the teachings of CHONG either anticipates or makes obvious the limitations and address the matters in the rejection.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lewis A. Bullock, Jr. whose telephone number is (571) 272-3759. The examiner can normally be reached on Monday-Friday, 8:30 a.m. - 5:00 p.m..

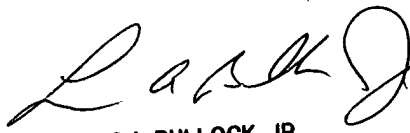
If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Application/Control Number:
09/867,650
Art Unit: 2195

Page 23

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

February 2, 2008



LEWIS A. BULLOCK, JR.
PRIMARY EXAMINER